ISBN: 978-628-95207-4-3. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2023.1.1.388

PENTEST in IoT Devices for a Precision agriculture Environment

Néstor X. Arreaga, Msig[1], Sara Blanc, PhD[2], Génesis Encalada[3], Eng and Miguel Neira, Eng[4]
[1,3,4] Escuela Superior Politécnica del Litoral, ESPOL, Polytechnic University, FIEC, Campus Gustavo Galindo Km. 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador, narreaga@espol.edu.ec, genmaenc@espol.edu.ec, migannei@espol.edu.ec
[1,2] ITACA, Universitat Politècnica de València, Valencia 46022, Valencia, España, sablacla@disca.upv.es

Abstact– Agriculture is constantly evolving, and many processes in the countryside have been automated owing to technology, giving rise to precision agriculture, which has helped to optimize the use of natural and agricultural inputs through the data sensed in the crops. Security in precision agriculture systems is vital to protect data; therefore, it has been proposed to perform penetration testing (PENTEST) or attacks on precision agriculture systems. In the experiment, open-source tools were used to determine vulnerabilities that may occur in real environments. Penetration tests were performed with Denial of Service, Man-in-the-Middle, and DNS spoofing attacks for devices on the Edge and in the Fog. These tests made it possible to affect the integrity, availability, and authenticity of the system. It was reported that service drops of more than 80% within the first minute, sensitive information theft, and page duplication allowed us to determine the vulnerability of this technology.

Keywords-- Precision agriculture, security, open source, vulnerabilities, Denial of Services.

I. Introduction

More information is being generated everyday, especially due to the applications of Internet of Things (IoT) solutions, [1]. With the emergence of appropriate devices, it is possible to obtain and process a large amount of data in a short time, creating unique systems that can be implemented in a wide range of scenarios. However, these devices may have security flaws in their applications, since they are new technology and are more vulnerable to leaks or attacks that can cause damage or data leaks in the system, [2].

On the other hand, among many IoT applications, there is precision agriculture (PA), which is in continuous development. PA provides great improvements in agricultural production, such as optimization of resources in precision irrigation of crops, decreasing the environmental impact, monitoring production in real time, thus maintaining the normalized difference vegetation index (NDVI) [3].

The implementation of PA depends on IoT devices and applications, which without proper security, are an easy target for a malicious attacker to access sensitive data, either for illegal reproduction or to negatively affect crops and their production. For this reason, a penetration test is performed on a precision agriculture application, which will allow to detect its vulnerabilities, and consequently to make preventive corrections, to develop in the future a robust system that will be difficult to penetrate.

We are living in an era in which agriculture is going through changes, there is a contrast in the techniques used for crops and, in addition, more developed countries have automated most of these processes, needing little human capital, [4]. On the other hand, poorer countries still perform the entire production chain manually, and developing countries are going through this transition in which precision agriculture is being incorporated into their crops.

IoT devices have had a great growth in this decade, due to the strong trend of making everything smart, also reaching the agricultural industry. One example is Central Africa, which has taken it upon itself to improve its agricultural production, ensuring that its crops are in optimal soil condition [5]. In the same way, agricultural companies have turned to implement these smart solutions because this allows them to increase their profit margin, but in most cases without considering safety issues.

Maintaining security breaches can cause an environment to be unsafe, and thus damage future crops, resulting in losses to the company's economy. Likewise, if the economy of an entire country depends on agriculture, it can be crucial if it is affected by agro-terrorism [6].

The ease with which an IoT device can be altered by some external agent, and it can perform erroneous activities, is high. Currently, there are AP systems that are exposed or vulnerable to any type of cyberattack, these attacks generate problems in maintaining control of the systems causing failures in their operation and making them vulnerable.

Small and medium-sized companies are the most affected by cyber-attacks because they are much more vulnerable. It has been estimated that since 2014, more than 60% of the attacks were made to this type of companies, among which were affected agricultural suppliers and companies dedicated to the agricultural sector, on the other hand, in 2015, about 75% of spear-phishing attacks [7] were directed to small businesses, among which were included farms.

This article contributes to evaluate the security of the precision agriculture system devices simulated through a

PENTEST using Open-Source tools. With the purpose of submitting it to an ethical hacking for the determination of the existing vulnerability gaps, as well as to elaborate a report that presents the result of the pentest through an analysis of variables such as availability, authenticity, and integrity of the data.

The structure of this document is as follows: Section II describes related work associated with penetration testing to validate security in different scenarios. Section III presents the performed methodology detailing the different tests. Section IV discusses the results observed by analyzing the attacks that have been performed on the system. Section V provides a discussion of the work presented. Finally, Section VI presents the conclusions and future work.

II. Related Work

IoT technologies have had a great acceptance, being one of the most advanced due to the available market. However, it has risks and are targets of attacks.

In [8], a PENTOS test system is proposed, which is used for IoT devices. It uses Kali Linux, one of the most common tools in ethical hacking, which is applied to perform different types of tests on the aforementioned devices. Following the basic security guide of OWASP (Open Web Application Security Project), finally the system provides the results of all modules and gives recommendations for security measures to be taken.

The authors in [9], propose a framework that analyzes the end-to-end penetration testing of an entire system that has IoT devices, it is possible to mimic the attacks that are provoked in a real way. The pentest verifies if the system has the necessary security measures. However, some of the pentests are not efficient because they test each device separately and not the entire system as a whole, unlike the system presented in this paper, which performs a pentest on the edge node, since it is the most vulnerable to external attacks.

In [2], it was proposed to perform tests in order to validate security in a modular way, so that the security level of each of them can be evaluated, for example, in the case of the web interface, tests were performed to protect against ClikJacking, vulnerability to SQL injection, vulnerability to XSS, among others. In the case of network services, they performed tests for resistance to denial-of-service attacks and fuzzing tests. For cryptography, they opted for an analysis of the protocols used. Then, in the firmware module, they performed readable word analysis, hexadecimal extraction, and extraction of the firmware file system.

Nowadays, most countries are migrating their infrastructure to 5G, but it has not been considered that M2M and IoT industrial communications use 2G and 3G networks in their already employed solutions. Due to the difficulty of migrating these solutions, some manufacturers propose to improve security in the devices, mostly used as Zigbee, GPS and LoRaWAN [10], as an alternative.

In [11], a Man in the Middle attack was deployed remotely to measure how vulnerable the created nodes were. The vulnerability analysis was performed using the multi-cell configuration of the OpenBTS open-source API for IoT nodes. As a result, even though there were unusual behaviors in the network and in the spectrum outside the established ranges, the GSM node was penetrated when one of its neighboring stations was spoofed. In our proposal we also performed a Man in The Middle attack through ARP Spoofing, as well as other types of attacks, such as DoS and DNS Spoofing, allowing us to know how the system is affected by different scenarios and vulnerabilities.

In the exploratory study in [12], they performed tests to detect vulnerabilities in seven different web applications, using pentesting tools (Htcap, SqlMap, Wapiti, XSSer, ZAP) to access the apps, considering the OWASP TOP10 project that lists the most common web vulnerabilities, being one of them the input data validation. Unlike our tests in which tools such as hping3, wireshark and bettercap were used, but with the same common goal.

In contrast, [13] is based on teaching and practice, where a CyExec platform was developed, using low-cost applications, being beneficial since most of the projects could not be implemented, either because of the high cost of deploying the equipment network or because of the virtuality due to the health situation, based on MV and Dockers containers. Also, reproducing them in virtual environments, using MV of Metasploitable2 and Kali Linux, and thus using the inspection tool to know the attack method.

III. Methodology

In order to perform the tests, it was necessary to implement a topology similar to the one that is used in precision agriculture, which has seven nodes communicating with each other wirelessly by radio frequency. It is possible that most of the areas or lands dedicated to agriculture do not have good coverage, for that reason, this communication option would be adapted to the environment.

An Arduino Uno board with an Atmega328p microcontroller [14] is used in each node, in addition to the sensors and actuators needed to perform the respective measurements required by the crop [15]. It is proposed that each of these nodes are communicated using the NRF24L02 radio frequency module [16], which operates in the free 2.4GHz band.

The deployment of seven nodes can cover an area of 25 hectares, according to the specifications of the module's transmission range. Of the total of seven nodes, only one will be the Gateway or edge node, which consisted of an Arduino and a Raspberry Pi. This node had Internet connectivity, for this it will need an Ethernet Shield that is coupled to the edge node, as shown in Fig. 1.

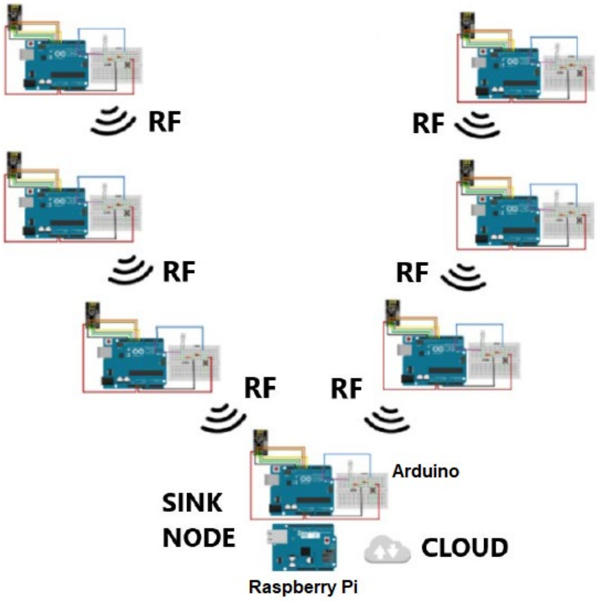


Fig. 1 Communication scheme between nodes

Penetration testing can be performed from several perspectives, attacking wireless communications, or attacking edge communication. In this article we will proceed to explain the pentest methodology for the edge node, since it is the one connected to the Internet and is more vulnerable to external attacks for radio frequency (RF) communications.

To perform the attack tests, several open-source platforms were analyzed, among which we have: Kali Linux [17, 18], which specializes in security auditing and is equipped with more than 300 tools for penetration testing, although this makes it slower than other distributions. On the other hand, Metasploit [19], which is easy to use, and has good documentation. It is scalable as you can create your own scripts, has about 800 attack scripts and can be run on different OS. It is also useful when system vulnerabilities are already known. Finally, Nessus [20], which has more than 80,000 configurable plugins for testing, has the option of port scanning and is easy to install.

After the analysis of its advantages and disadvantages, Kali Linux was chosen over the others because it does not need a client-server architecture, and it is not necessary to learn its own language like Nessus. For the tests we used a Windows computer as a host where a Kali Linux virtual machine is running inside VirtualBox, making Kali as a guest. The different Arduino devices [21] were connected to the host machine. The Kali console is enough to execute the necessary commands to perform the tests to the devices.

A. Implementation Design

To connect the Arduino with the internet, we attached an Ethernet shield or a wifi module to the board (except for the Raspberry, since it has a built-in wifi module) [22]. The Arduino was connected via ethernet cable to the router or access point that has the network in which we are going to work. The laptop was connected via wireless, which made it possible to be on the same network segment. The card was programmed to transmit the readings of its analog inputs to the internet, and the IP addresses of each device were searched to establish the connection between the computer and the sensor nodes (Arduino).

Kali Linux was installed in the VirtualBox software, the Kali virtual machine was executed and the commands to perform the different tests were run in its console, this can be seen diagrammed in Fig. 2.

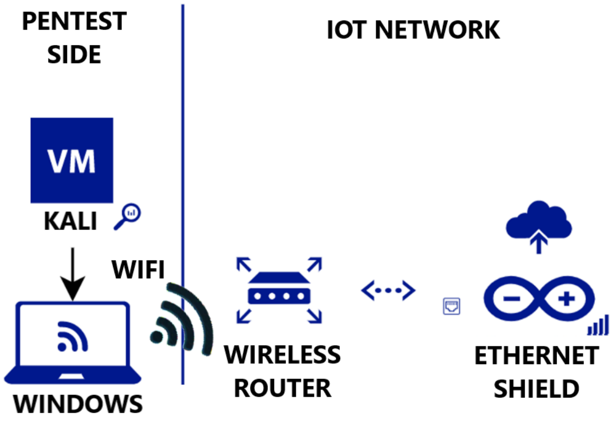


Fig. 2 Schematic of the pentest to the Arduino node

There are layers in the processing of data collected in IoT environments. Most authors divide them into three: Edge, Fog and Cloud, these layers were originated from the need to analyze information, being the Edge layer the most exposed.

In the Edge layer are located the sensors and actuators of the system, sometimes this layer can also process data, but at a very low scale [23].

Above the Edge layer is the Fog layer and depending on the amount of data it can be processed at local nodes. In this layer, data is not directly uploaded to the cloud (next layer), but can be processed internally, it is intended to act quickly and immediately. On the other hand, there is the Cloud layer, which performs the computing in the cloud. This last layer has a great processing and will depend on what service is being used, in the market there are several recognized options such as Ubidots, Microsoft Azure, Google's IoT Core, AWS IoT, among others.

B. *DoS Attack Test*

For the first test, we performed a denial-of-service attack. For this, a Kali virtual machine was introduced to the network. This was done using the bridge adapter network configuration, allowing to take addressing within the local network and making it easier to send packets to the network, [24].

In order to limit the transmission made by the Arduino, we used the Hping3 tool, [25], which helps the analysis and assembly of TCP/IP packets, similar to Windows ping that only allows sending ICMP packets. This allows specifying the number of packets, size, flag type, among others. We also use Wireshark in conjunction with the previous tool to analyze the packets.

- Before DoS Attack

  The page is functional and transmitting, since the command has not yet been executed for the DoS attack to occur, a response time is obtained in the command prompt window by pinging the server address. It is observed that the packets have been sent and received successfully and without loss.

- After DoS Attack

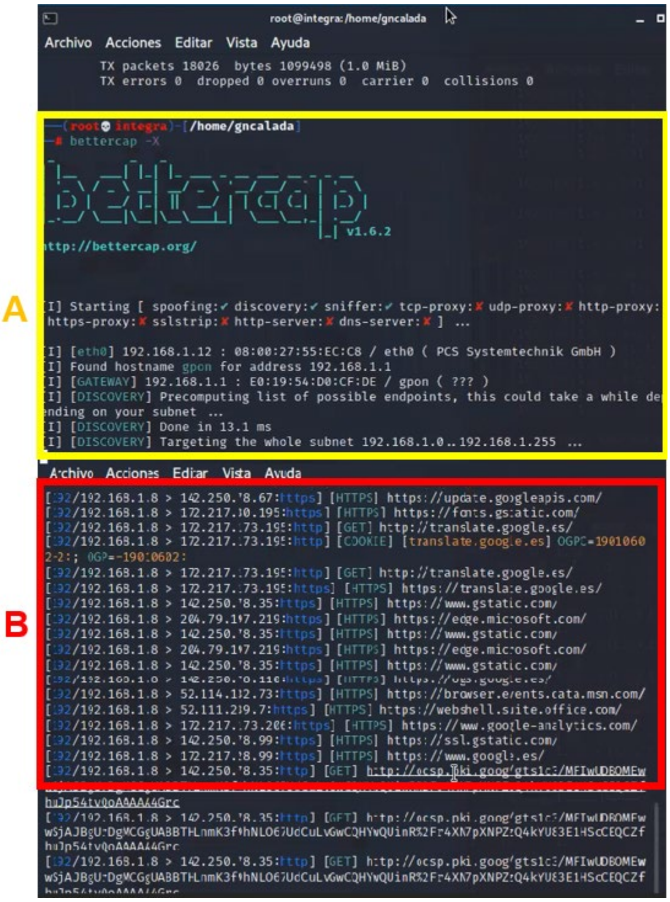
  After executing the command line in the Shell, the attack is generated, in this case, in the browser window the message "Your connection was interrupted" will appear. In the command prompt window, when establishing connectivity using the ping command, the window will display "Destionation Host Unreachable", which proves that the server is not responding, i.e., a connection cannot be established, requests are sent, but as network traffic is saturated, they are lost.

C. *ARP Spoofing – Man In The Middle Attack Test*

As can be seen in Fig. 3, the Bettercap tool is being used to perform the attack, since all the packets that should pass through the Gateway are passing through the Kali device to go out to the network. Thus, all the computers connected to the network that make queries will appear in the panel with the type of request they have made.

This can be verified in the red framed section (B), where it is possible to observe the queries with the IP addresses and the pages where the information is being sent. For this it is necessary to configure the Kali equipment as part of the local network, in the Kali VM configurations it was given a local address.

When starting to scan all the devices in the network, including the transmission that the Arduino performs with the measurements, this can be seen in the yellow framed section (A), it is possible to listen to the information that the IoT devices talk to each other, to be able to do this, the bettercap -x command is used and the tool is executed, at that moment it listens to the devices at the same time.


root@integra: /home/gncalada
Archivo Acciones Editar Vista Ayuda
A
Archivo Acciones Editar Vista Ayuda
B

```
        TX packets 18026  bytes 1099498 (1.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(root💀integra)-[/home/gncalada]
└─# bettercap -X

                                              v1.6.2
http://bettercap.org/


[I] Starting [ spoofing:✓ discovery:✓ sniffer:✓ tcp-proxy:✘ udp-proxy:✘ http-proxy:
https-proxy:✘ sslstrip:✘ http-server:✘ dns-server:✘ ] ...

[I] [eth0] 192.168.1.12 : 08:00:27:55:EC:C8 / eth0 ( PCS Systemtechnik GmbH )
[I] Found hostname gpon for address 192.168.1.1
[I] [GATEWAY] 192.168.1.1 : E0:19:54:D0:CF:DE / gpon ( ??? )
[I] [DISCOVERY] Precomputing list of possible endpoints, this could take a while de
pending on your subnet ...
[I] [DISCOVERY] Done in 13.1 ms
[I] [DISCOVERY] Targeting the whole subnet 192.168.1.0..192.168.1.255 ...
```

```
[192/192.168.1.8 > 142.250.78.67:https] [HTTPS] https://update.googleapis.com/
[192/192.168.1.8 > 172.217.30.195:https] [HTTPS] https://fonts.gstatic.com/
[192/192.168.1.8 > 172.217.173.195:http] [GET] http://translate.google.es/
[192/192.168.1.8 > 172.217.173.195:http] [COOKIE] [translate.google.es] OGPC=1901060
2-2:; OGP=-1900602:
[192/192.168.1.8 > 172.217.173.195:http] [GET] http://translate.google.es/
[192/192.168.1.8 > 172.217.173.195:https] [HTTPS] https://translate.google.es/
[192/192.168.1.8 > 142.250.78.35:https] [HTTPS] https://www.gstatic.com/
[192/192.168.1.8 > 204.79.197.219:https] [HTTPS] https://edge.microsoft.com/
[192/192.168.1.8 > 142.250.78.35:https] [HTTPS] https://www.gstatic.com/
[192/192.168.1.8 > 204.79.197.219:https] [HTTPS] https://edge.microsoft.com/
[192/192.168.1.8 > 142.250.78.35:https] [HTTPS] https://www.gstatic.com/
[192/192.168.1.8 > 142.250.78.110:https] [HTTPS] https://ogs.google.es/
[192/192.168.1.8 > 52.114.132.73:https] [HTTPS] https://browser.events.data.msn.com/
[192/192.168.1.8 > 52.111.239.7:https] [HTTPS] https://webshell.suite.office.com/
[192/192.168.1.8 > 172.217.173.206:https] [HTTPS] https://www.google-analytics.com/
[192/192.168.1.8 > 142.250.78.99:https] [HTTPS] https://ssl.gstatic.com/
[192/192.168.1.8 > 172.217.28.99:https] [HTTPS] https://www.google.es/
[192/192.168.1.8 > 142.250.78.35:http] [GET] http://ocsp.pki.goog/gts1c3/MFIwUDBOMEw
huJp54tvQoAAAAA4Grc
[192/192.168.1.8 > 142.250.78.35:http] [GET] http://ocsp.pki.goog/gts1c3/MFIwUDBOMEw
wSjAJBgUrDgMCGgUABBTHLnmK3f9hNLO67UdCuLvGwCQHYwQUinR%2Fr4XN7pXNPZzQ4kYU83E1HSCCEQCZf
huJp54tvQoAAAAA4Grc
[192/192.168.1.8 > 142.250.78.35:http] [GET] http://ocsp.pki.goog/gts1c3/MFIwUDBOMEw
wSjAJBgUrDgMCGgUABBTHLnmK3f9hNLO67UdCuLvGwCQHYwQUinR%2Fr4XN7pXNPZzQ4kYU83E1HSCCEQCZf
huJp54tvQoAAAAA4Grc
```

Fig. 3 During MiTM

D. *DNS Spoofing Attack Test*

This request to the server can be used to redirect the query. This is how DNS spoofing makes it possible to elaborate the third scenario, which allows that when an IP address is queried, a different page is displayed (if that is the objective) to the one who made the query.

In this case, it is proposed that this attack will be passive, meaning that the changes to the page will be imperceptible. Thus, those who use it will not notice that they have been breached, but rather will believe that the data being presented are reliable. The diagram in Fig. 4 explains how the attack was carried out.

An apache server is raised the same with respect to the measured data where the HTML code matches the original page, since the page was configured so that initially the transmission is updated every five seconds. This is done to avoid any doubt that the original site is being visited, the only difference is the data presented, since that data is not correct.

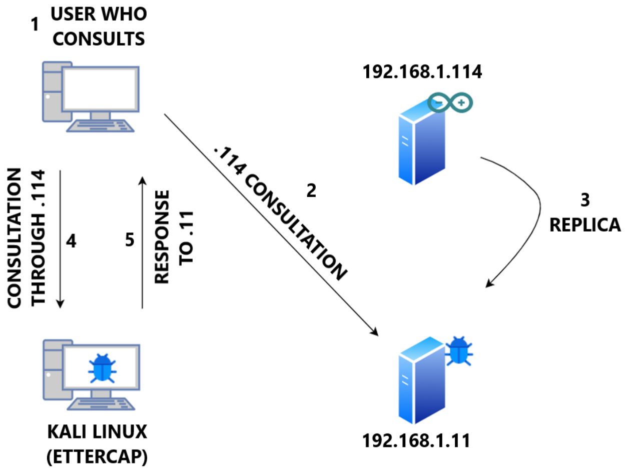


Fig. 4 DNS Spoofing attack scheme

It is possible to experiment with the values by presenting very high or low values depending on the measurements, if the values were subject to actuators, they would cause serious physical damage. Since, if the humidity is set to a very low value, irrigation pumps are activated and consequently crops can be drowned.

IV. Results and Analysis

DoS, MITM and DNS Spoofing tests were performed on the Arduino at the Edge level, and on the Raspberry at the Fog level [26]. In both layers several samples were taken in order to study the results.

A. *DoS Attack*

This attack was executed in two scenarios, in the first one the Edge layer is evaluated with the Gateway node connected by Ethernet cable to the access point. On the other hand, connected via Wifi through the Esp8266 module wirelessly at approximately 10 meters.

Both Table 1 and Table 2 show the results of the DoS at the Edge layer. For this attack, we used the hping3 tool where four attempts were made, varying only the number of packets sent. It was kept the TCP transmission protocol and all the packets were of SYN type with a size of 1200 bytes with random source addresses, to hide the attacker's address Kali (192.168.1.11). It is possible to observe the mentioned above because Wireshark was running in the background analyzing the sent packet burst.

TABLE I
Results of the DoS Attack on the Edge Layer

Experiment Number	Number of Packages	Drop Time	Packet Loss
1	500	40s	63%
2	700	22s	87%
3	1000	6s	98%
4	1300	2s	100%

TABLE II
Results of the DoS Attack on the Edge Layer Wirelessly (10 meters)

Experiment Number	Number of Packages	Drop Time	Packet Loss
1	500	47s	61%
2	700	25s	80%
3	1000	7s	95%
4	1300	2s	100%

There are few differences that can be noticed between performing the attack via wireless or directly wired to an access point. These differences are the speed with which the service is dropped, and the percentage of packets lost.

Although in the experiment this difference is very small, what stands out the most is that since it is a wired medium, when a medium-low burst is sent, there is a difference of between 3 to 7 points with respect to the percentage of packets lost. However, when large bursts are sent, they tend to give a similar drop as if they were in a wireless medium, as can be seen in the comparative graph shown in Fig. 5.

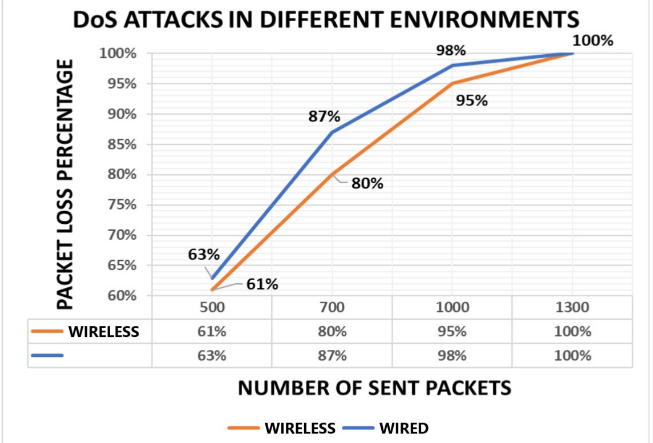


Fig. 5 Comparative chart between DoS attack environments

On the other hand, Table 3 shows the results of the DoS attack performed on the Fog side, for this, we use a Raspbian, which is the O.S. of the Raspberry Pi (where our server would be installed), and to which the data would be sent when it is collected to be processed locally.

TABLE III
RESULTADOS OF THE DOS ATTACK ON THE FOG LAYER

Experiment Number	Number of Packages	Drop Time	Packet Loss
1	800	8min 5s	75%
2	1000	7min 13s	81%
3	1300	5min 47s	93%
4	1500	3min 6s	99%

In this experiment we used Metasploit to perform the DoS attack. The attack was sent through the attacking Kali machine towards the address of the previously raised Apache server. In this case it was necessary to have three super user terminals running at the same time. In addition, in each of them the attack is running, in order to find significant results such as delays in ping times to the server.

During the first few minutes it was imperceptible that it was running an attack, and even more so since the number of packets was very small; once enough packets were sent, the server was disabled and a successful DoS to the Fog occurred in approximately three minutes.

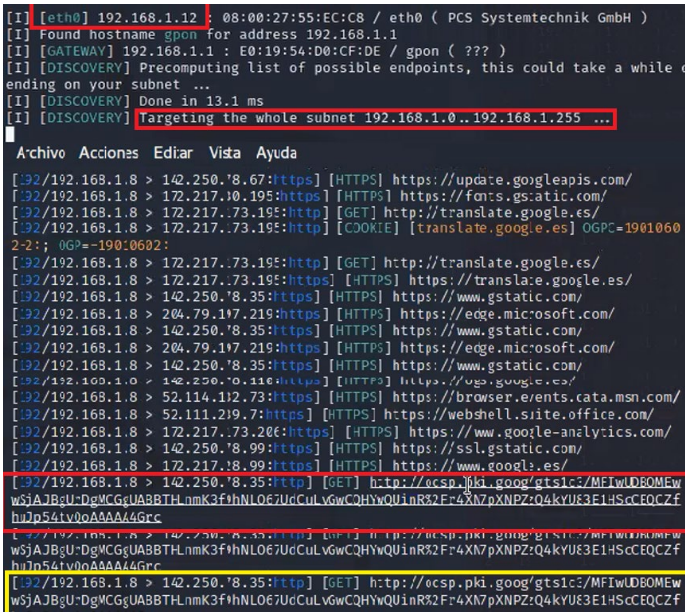
B. *ARP Spoofing – Man In The Middle Attack*

This type of attack was carried out passively to prevent anyone from realizing that they were being "eavesdropped". Thus, the system can work normally and intercept the necessary information. Similarly, the attack was performed on both layers, Edge and Fog, but using different tools: Bettercap was used for Edge and Ettercap for Fog.

In the Edge, via a command line, we invoked the Bettercap tool as shown in Fig. 6. In this way, a sniffer is made together with WireShark, through a scan of the IP and MAC addresses of the hosts on the network to identify the victim and the destination of the packets. Then they were sent ARP packets with the MAC of the Kali machine (attacker) and the IP of the victim (Arduino transmitting).

In this way, all other hosts on the network that are sending information to the victim are redirected to the Kali attacker, who is in listening mode over every movement made by the network, including transmissions. On the other hand, in fog, before starting the attack, it is necessary to scan the network and configure the targets or IPs of the victim and the attacker. Fig. 7 shows how it is working together with WireShark, where the attack is already up and running.

From the Raspbery Pi, which acts as a server, a Ping is sent to another host on the network. This ICMP communication is intercepted by the Kali attacker using the same mechanics explained previously for the Edge. The internet was browsed from the Raspbian OS and HTTP communications were intercepted. In this way, sensitive data such as users and passwords on pages that still continue to use this protocol were obtained.

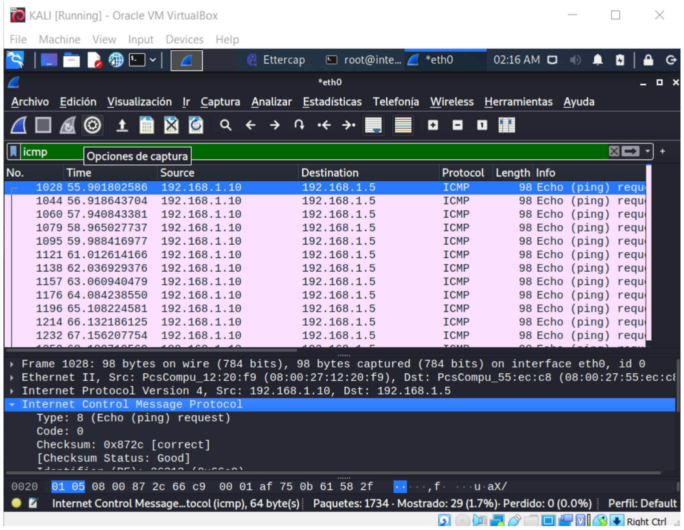


```
[I] [eth0] 192.168.1.12 : 08:00:27:55:EC:C8 / eth0 ( PCS Systemtechnik GmbH )
[I] Found hostname gpon for address 192.168.1.1
[I] [GATEWAY] 192.168.1.1 : E0:19:54:D0:CF:DE / gpon ( ??? )
[I] [DISCOVERY] Precomputing list of possible endpoints, this could take a while d
ending on your subnet ...
[I] [DISCOVERY] Done in 13.1 ms
[I] [DISCOVERY] Targeting the whole subnet 192.168.1.0..192.168.1.255 ...
```

```
[192/192.168.1.8 > 142.250.78.67:https] [HTTPS] https://update.googleapis.com/
[192/192.168.1.8 > 172.217.30.195:https] [HTTPS] https://fonts.gstatic.com/
[192/192.168.1.8 > 172.217.173.195:http] [GET] http://translate.google.es/
[192/192.168.1.8 > 172.217.173.195:http] [COOKIE] [translate.google.es] OGPC=1901060
2-2:; OGP=-19010602:
[192/192.168.1.8 > 172.217.173.195:http] [GET] http://translate.google.es/
[192/192.168.1.8 > 172.217.173.195:https] [HTTPS] https://translate.google.es/
[192/192.168.1.8 > 142.250.78.35:https] [HTTPS] https://www.gstatic.com/
[192/192.168.1.8 > 204.79.197.219:https] [HTTPS] https://edge.microsoft.com/
[192/192.168.1.8 > 142.250.78.35:https] [HTTPS] https://www.gstatic.com/
[192/192.168.1.8 > 204.79.197.219:https] [HTTPS] https://edge.microsoft.com/
[192/192.168.1.8 > 142.250.78.35:https] [HTTPS] https://www.gstatic.com/
[192/192.168.1.8 > 142.250.78.110:https] [HTTPS] https://ogs.google.es/
[192/192.168.1.8 > 52.114.132.73:https] [HTTPS] https://browser.events.cata.msn.com/
[192/192.168.1.8 > 52.111.239.7:https] [HTTPS] https://webshell.suite.office.com/
[192/192.168.1.8 > 172.217.173.206:https] [HTTPS] https://www.google-analytics.com/
[192/192.168.1.8 > 142.250.78.99:https] [HTTPS] https://ssl.gstatic.com/
[192/192.168.1.8 > 172.217.28.99:https] [HTTPS] https://www.google.es/
[192/192.168.1.8 > 142.250.78.35:http] [GET] http://ocsp.pki.goog/gts1c3/MFIwUDBOMEw
wSjAJBgUrDgMCGgUABBTHLnmK3f9hNLO67UdCuLvGwCQHYwQUinR%2Fr4XN7pXNPZzQ4kYU83E1HScCEQCZf
huJp54tvQoAAAA44Grc
[192/192.168.1.8 > 142.250.78.35:http] [GET] http://ocsp.pki.goog/gts1c3/MFIwUDBOMEw
wSjAJBgUrDgMCGgUABBTHLnmK3f9hNLO67UdCuLvGwCQHYwQUinR%2Fr4XN7pXNPZzQ4kYU83E1HScCEQCZf
huJp54tvQoAAAA44Grc
[192/192.168.1.8 > 142.250.78.35:http] [GET] http://ocsp.pki.goog/gts1c3/MFIwUDBOMEw
wSjAJBgUrDgMCGgUABBTHLnmK3f9hNLO67UdCuLvGwCQHYwQUinR%2Fr4XN7pXNPZzQ4kYU83E1HScCEQCZf
```

Fig. 6 MiTM attack on the Edge layer using Bettercap

Both for this and the other case of attack, no time is considered, since the objective of a MITM is to act passively and be almost imperceptible to others, which was achieved through these attacks.



KALI [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Ettercap
root@inte...
*eth0
02:16 AM
*eth0
Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda
Opciones de captura
Right Ctrl

Filter: icmp

```
No.  Time         Source        Destination  Protocol Length Info
1028 55.901802586 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1044 56.918643704 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1060 57.940843381 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1079 58.965027737 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1095 59.988416977 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1121 61.012614166 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1138 62.036929376 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1157 63.060940479 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1176 64.084238550 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1196 65.108224581 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1214 66.132186125 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ
1232 67.156207754 192.168.1.10  192.168.1.5  ICMP     98 Echo (ping) requ

› Frame 1028: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu_12:20:f9 (08:00:27:12:20:f9), Dst: PcsCompu_55:ec:c8 (08:00:27:55:ec:c8
› Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.5
˅ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x872c [correct]
    [Checksum Status: Good]

0020  01 05 08 00 87 2c 66 c9  00 01 af 75 0b 61 58 2f   ·····,f· ···u·aX/

Internet Control Message...tocol (icmp), 64 byte(s) | Paquetes: 1734 · Mostrado: 29 (1.7%)· Perdido: 0 (0.0%) | Perfil: Default
```

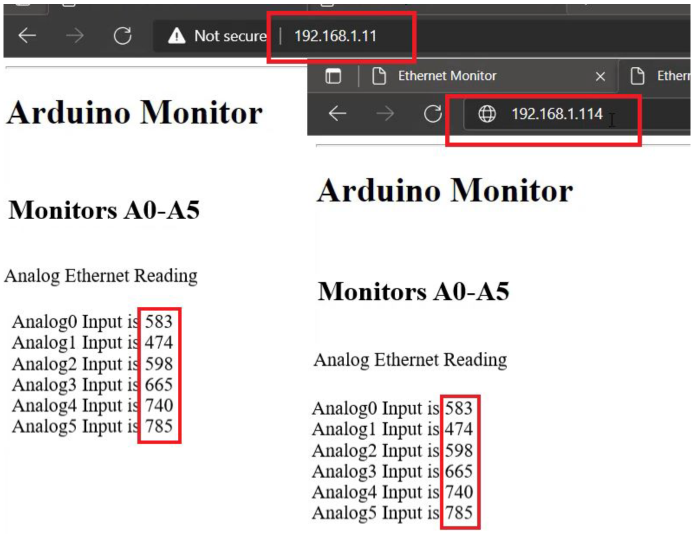
Fig. 7 MiTM attack on the Fog layer using Ettercap

C. *Ataque DNS Spoofing*

The objective of this type of attack depends on whether you want to perform an ethical hack or not. In this project we were looking to replicate exactly the page that monitors the data sent by the Arduino, and then sent to the server. It was performed a scenario applied to Precision Agriculture, showing a page where the data from the sensors in the field is presented in real time and displays the information obtained from the Edge, or otherwise from the Fog in case the information is more processed [27].

For this attack we also used Ettercap in both layers, we set up an apache server from which we changed the main HTML page to the Arduino monitor. A rule was set up in /var/www/html from the Kali VM where the IP addresses are switcheted, and from Ettercap we ran DNS spoofing for both the Edge and Fog. The same steps were performed to redirect the IP addresses presented by the Arduino monitor to the page that was already pre-created with information like the original. In this way giving the impression that nothing has happened, the difference will be the values and that you must also have prior knowledge of what the page looks like.

In Fig. 8 it can be seen how it has been set that the address .114 of the Arduino, where the updates of the measurements are sent, is the same as the server raised in .11, being almost imperceptible, but if changes are made in .11, .114 will also change and this will cause inconsistency in the data.



Arduino Monitor

Monitors A0-A5

Analog Ethernet Reading

Analog0 Input is 583
Analog1 Input is 474
Analog2 Input is 598
Analog3 Input is 665
Analog4 Input is 740
Analog5 Input is 785

Arduino Monitor

Monitors A0-A5

Analog Ethernet Reading

Analog0 Input is 583
Analog1 Input is 474
Analog2 Input is 598
Analog3 Input is 665
Analog4 Input is 740
Analog5 Input is 785

Fig. 8 DNS Spoofing Attack at the Fog layer with Bettercap

The DoS test is considered to be active; its layers of affection are Edge and Fog, it presents consequences related to the server being unavailable for queries, visualization and activation of components, delay of watering and fertilization times, thus violating the security aspect of availability.

The layers of Man-in-the-Middle (MitM) testing are Edge and Fog, and the consequences presented are theft of sensitive information, corporate secrets, private cultivation techniques, mixing of inputs that increase the yield of the fields, affecting security aspects such as integrity, privacy, and confidentiality.

Finally, the DNS Spoofing test is a passive type, whose affection layer is Fog and has consequences such as the presentation of false information, sabotage in the activation of components such as irrigation, where crops can be spoiled by presenting information that is not favorable to the company. In addition, users and passwords can be stolen from the system if phishing techniques are used, affecting security aspects such as integrity and authenticity.

V. Discussion

After performing the denial of service test in the Edge layer by wired connection, and using the Hping3 tool, it was observed that by increasing the rate of packets sent, the system collapses proportionally, which caused the user to have no access to the service due to the server drop, this caused the drop time to be approximately 47 seconds, and that out of every 500 packets, 305 are lost, having an average of 61% of packets lost in the experiment.

The DoS test performed on the Fog layer using an exploit took 3 minutes and 6 seconds to get 99% of the 1500 packets sent to be lost, compared to the 2 seconds it takes using hping3 on the Edge layer. If we compare the time that the server takes to drop in the tests, it is possible to observe that when the number of packets sent is small, for example, 500 packets, the drop time is further separated according to the medium in which the attack occurs, i.e., by direct connection the packet loss of 63% in 40 seconds.

On the other hand, the wireless test has a packet loss of 61% after 47 seconds with the same number of packets. The fourth experiment, in contrast, shows a loss of 100% with 1300 packets after 2 seconds for both ways, which means that, regardless of the method by which an attacker connects, he can still leave the system without service, but there will be a difference depending on the number of packets and the method used.

In the MitM test, the capture of the data obtained through the Wireshark sniffer allowed us to know the transmission of the Edge node, while we could filter by protocols of interest and know the transmission frequency of the node.

VI. Conclusions

Through the DoS attack performed on both the Edge and Fog layers, it was concluded that the hping3 tool is more effective than an exploit, and the fog layer is more robust for processing sensitive data information. However, this test can be improved by using more machines trying to attack the same server (DDoS), and the vulnerabilities of the system can be further reduced.

Because MiTM and DNS Spoofing attacks are passive, there is no evident drop-in service; however, the information continues to be processed and transmitted.

Furthermore, the use of Wireshark for the MiTM test provided information that could be used to develop further attacks, replicate, or clone the node's information. In addition to the fact that the data handled were exposed much more, it was possible to listen to the packets sent from the network to the Internet through the HTTP protocol.

In addition, it is concluded that this technology is very vulnerable because a significant number of service failures were reported in a very short period; therefore, it is recommended to create an intrusion detection system (IDS) or an intrusion prevention system (IPS) to protect the precision agriculture network from possible cyber-attacks.

It is also important to mention that with the increasing reliance on technology in agriculture, it is essential to ensure the security of these systems to protect the productivity and reliability of farming operations by performing regular penetration tests to detect any possible malicious attacks and to avoid both economic and security losses in the system.

For this reason, we were able to successfully evaluate the security of a precision agriculture system simulated through pentests using open-source tools that allowed us to determine the existing vulnerability gaps.

For future work, we intend to perform a pentest with other types of attacks such as DNS cache poisoning, IP spoofing, KeyLoggers, and TCP sequence attacks. Likewise, implementing defense systems, such as an IDS/IPS, re-validates the test, and determines the efficiency of incorporating defense techniques against security threats. Additionally, it creates a topology with more sensor nodes and requirements in the field of agriculture.

References

[1] A. M. Awadelkarim Mohamed and Y. Abdallah M. Hamad, “IoT Security: Review and Future Directions for Protection Models,” 2020 Int. Conf. Comput. Inf. Technol. ICCIT 2020, pp. 166–169.

[2] C. Cristoffer, J. J. Gondim, P. S. Barreto, M. F. Caetano, and E. A. Alchieri, “Pentest on internet of things devices,” Proc. - 2019 45th Lat. Am. Comput. Conf. CLEI 2019, 2019.

[3] N. J. Chapungo and O. Postolache, “Sensors and Comunication Protocols for Precision Agriculture,” pp. 1–6, 2021.

[4] R. Gebbers and V. I. Adamchuk, “Precision agriculture and food security,” Science, vol. 327, no. 5967, pp. 828–831, Feb. 2010.

[5] S. Rizvi, N. McIntyre, and J. Ryoo, “Computing Security Scores for IoT Device Vulnerabilities,” Proc. - 2019 Int. Conf. Softw. Secur. Assur. ICSSA 2019, pp. 52–59, 2019.

[6] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, “Security and Privacy in Smart Farming: Challenges and Opportunities,” IEEE Access, vol. 8, pp. 34564–34584, 2020.

[7] J. West, “A Prediction Model Framework for Cyber-Attacks to Precision Agriculture Technologies,” J. Agric. Food Inf., vol. 19, no. 4, pp. 307–330, 2018.

[8] V. Visoottiviseth, P. Akarasiriwong, S. Chaiyasart, and S. Chotivatunyu, “PENTOS: Penetration testing tool for Internet of Thing devices,” IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 2017-December, pp. 2279–2284, 2017.

[9] G. Yadav, A. Allakany, V. Kumar, K. Paul, and K. Okamura, “Penetration Testing Framework for IoT,” Proc. - 2019 8th Int. Congr. Adv. Appl. Informatics, IIAI-AAI 2019, pp. 477–482, 2019.

[10] A. Griva, A. D. Boursianis, S. Wan, P. Sarigiannidis, G. Karagiannidis, and S. K. Goudos, “Performance Evaluation of LoRa Networks in an Open Field Cultivation Scenario,” 2021 10th Int. Conf. Mod. Circuits Syst. Technol. MOCAST 2021, pp. 1–5, 2021.

[11] J. De Jesus Rugeles Uribe and E. P. Guillen, “Vulnerability Assessment for IoT Nodes Using OpenBTS and Software Defined Radios,” 2020 3rd Int. Conf. Signal Process. Inf. Secur. ICSPIS 2020, pp. 3–6, 2020.

[12] L. F. De Lima, M. C. Horsgtmann, D. N. Neto, A. R. A. Grégio, F. Silva, and L. M. Peres, “On the Challenges of Automated Testing of Web Vulnerabilities,” Proc. Work. Enabling Technol. Infrastruct. Collab. Enterp. WETICE, vol. 2020-Septe, pp. 203–206, 2020.

[13] S. Shin and Y. Seto, “Development of IoT security exercise contents for cyber security exercise system,” Int. Conf. Hum. Syst. Interact. HSI, vol. 2020-June, pp. 281–286, 2020.

[14] K. Tsukada, M. Oki, T. Yamamoto, and T. Imaizumi, “CARduino: Device toolkit suitable for use in automobiles,” UbiComp ISWC 2015 - Proc. 2015 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Proc. 2015 ACM Int. Symp. Wearable Comput., vol. 148, pp. 407–410, 2015.

[15] A. Nayyar and V. Puri, “A review of Arduino board’s, Lilypad’s & Arduino shields,” in Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016, 2016, pp. 1485–1492.

[16] “NRF24L01 Datasheet(PDF) - List of Unclassifed Manufacturers.” https://www.alldatasheet.com/datasheet-pdf/pdf/1243924/ETC1/NRF24L01.html (accessed Sep. 04, 2021).

[17] R. T. Gaddam and M. Nandhini, “An analysis of various snort based techniques to detect and prevent intrusions in networks: Proposal with code refactoring snort tool in Kali Linux environment,” Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2017, no. Icicct, pp. 10–15, 2017.

[18] T. S. Gunawan, M. K. Lim, N. F. Zulkurnain, and M. Kartiwi, “On the review and setup of security audit using Kali Linux,” Indones. J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 51–59, 2018.

[19] H. Gupta and R. Kumar, “Protection against penetration attacks using Metasploit,” 2015 4th Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2015, pp. 2–5, 2015.

[20] R. Williams, E. McMahon, S. Samtani, M. Patton, and H. Chen, “Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach,” 2017 IEEE Int. Conf. Intell. Secur. Informatics Secur. Big Data, ISI 2017, pp. 179–181, 2017.

[21] O. E. Amestica, P. E. Melin, C. R. Duran-Faundez, and G. R. Lagos, “An Experimental Comparison of Arduino IDE Compatible Platforms for Digital Control and Data Acquisition Applications,” IEEE Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol. CHILECON 2019, pp. 1–6, 2019.

[22] N. S. Yamanoor and S. Yamanoor, “High quality, low cost education with the Raspberry Pi,” 2017.

[23] S. Blanc, J.-L. Bayo-Montón, S. Palanca-Barrio, and N. X. Arreaga-Alvarado, “A Service Discovery Solution for Edge Choreography-Based Distributed Embedded Systems,” Sensors, vol. 21, no. 2, p. 672.

[24] Y. Jiang, K. F. Zheng, Y. X. Yang, S. S. Luo, and J. P. Zhao, “Evaluation model for DoS attack effect in softswitch network,” Proc. - 2010 Int. Conf. Commun. Intell. Inf. Secur. ICCIIS 2010, pp. 88–91, 2010.

[25] A. C. Bovy, H. Chalopin, G. Bas, and P. Courmontagne, “IR-UWB: An high speed digital receiver for very short range transmissions,” 2008 IEEE Radio Wirel. Symp. RWS, pp. 363–366, 2008.

[26] E. Al Neyadi, S. Al Shehhi, A. Al Shehhi, N. Al Hashimi, M. Qbea’H, and S. Alrabaee, “Discovering Public Wi-Fi Vulnerabilities Using Raspberry pi and Kali Linux,” Proc. - 2020 12th Annu. Undergrad. Res. Conf. Appl. Comput. URC 2020, 2020.

[27] V. Khatod and A. Manolova, “Effects of Man in the Middle (MITM) Attack on Bit Error Rate of Bluetooth System,” 2020 Jt. Int. Conf. Digit. Arts, Media Technol. with ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng. ECTI DAMT NCON 2020, pp. 153–157, 2020.